LA TUA RIVISTA UFFICIALE DI RASPBERRY PI

The MagPi

La rivista ufficiale Raspberry Pi
in italiano, da RaspberryItaly.com

Numero 46 Giugno 2016
www.raspberryitaly.com



ELETTRONICA
CON IL RASPBERRY PI

Realizza i tuoi primi circuiti e progetti in semplici passi

The MagPi magazine is published by Raspberry Pi (Trading) Ltd., Mount Pleasant House, Cambridge, CB3 0RN. ISSN: 2051-9982

GUIDA PER Principianti IN Elett CON Raspberry Pi

onica

Usa i pin GPIO, qualche componente elettronico, e del codice per fare grandi cose con Raspberry Pi

Ogni mese su *The MagPi*, abbiamo le pagine piene di progetti che si collegano direttamente al Raspberry Pi utilizzando i pin GPIO che si trovano sul lato della scheda. Sono abbastanza semplici da utilizzare e, con un poco di pazienza, si può scoprire come funzionano leggendo tutti i vari tutorial.

Con la nascita della libreria Phynton GPIO Zero, non è mai stato così semplice iniziare a scrivere codice per dei progetti elettronici. Con questo bene in mente, abbiamo deciso di rimuovere tutti i passaggi che portano dall' idea di un progetto al suo sviluppo definitivo, ma di proporre una vera e propria guida su come far funzionare l'elettronica con il Raspberry Pi.

In questo estratto tradotto, ti spiegheremo esattamente come funzionano i vari componenti elettronici e come collegarli tra di loro, oltre ad alcuni cenni su come programmarli tramite GPIO Zero, prima di illustrare alcuni esempi significativi del tipo di cose che si possono realizzare con il Pi una volta che si conosce come procedere.

Tutto il codice in questo tutorial è Python 3

Iniziare con Raspberry Pi e l'Elettronica

Leggere gli schemi circuitali e cablarli con il tuo Raspberry Pi

Il Raspberry Pi è ottimo per imparare il mondo dell'informatica. Che si tratti di scrivere del codice, piuttosto che di un utilizzo più avanzato del computer, Raspberry Pi dispone di svariati strumenti che aiutano nell'apprendimento.

Fig 1 Circuito "interrotta"



È anche molto buono per interfacciarsi con il mondo fisico, che in questo contesto significa saper programmare ed interagire con il mondo reale utilizzando l'elettronica. In parole semplici, il physical computing con il Pi è ad esempio pilotare l accensione di un LED, un componente elettronico in un circuito elettronico.

I circuiti elettronici sono la parte fisica di un progetto di physical computing connesso al Raspberry Pi. Questi circuiti possono essere semplici o anche molto complessi, e consistono di componenti elettronici come il LED appena citato, cicalini, tasti, resistenze, condensatori, ed anche circuiti integrati (IC) detti "chip".

Nel caso più semplice, un circuito elettronico consente di far arrivare corrente ad alcuni componenti secondo un ordine specifico, fluendo dal terminale positivo del circuito, fino al negativo (o massa). Pensa alla lampadina di casa: la corrente ci passa attraverso, e quindi la accende. Puoi aggiungere un interruttore che interrompe il circuito e così puoi accendere la luce quando lo chiudi, e spegnerla quando apri il circuito. Ecco: questo è un circuito elettronico.

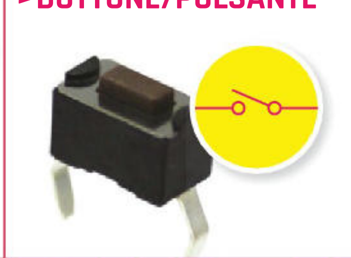
Leggere gli schemi

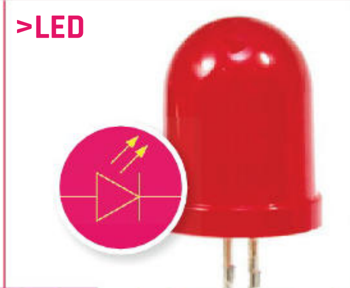
Progettare un circuito può risultare facile se sai cosa stai facendo, ma se stai sperimentando un nuovo circuito, o se sei nuovo nel campo dell'elettronica in generale, devi fare riferimento ad uno schema del circuito stesso. Questo è il modo usuale per rappresentare un circuito e gli schemi risultano molto più semplici da leggere e capire rispetto a una fotografia del circuito. Tuttavia, i componenti sono rappresentati con dei simboli che devi conoscere, ed è quindi necessario impararli.

La Fig 1 è un esempio del circuito di accensione di un LED a cui si faceva cenno in precedenza. In questo schema abbiamo una fonte di energia (una batteria), un interruttore, una resistenza, ed un LED. Le linee rappresentano come i

SIMBOLI DEI COMPONENTI PIU' COMUNI









Credit: oomlout

Questi jumper o"cavallotti" di filo sono molto utili per connettersi ai pin GPIO



circuiti sono connessi tra loro con del filo o con altri metodi. Alcuni componenti possono essere collegati senza uno specifico verso, come le resistenze o gli interruttori. Altri invece hanno un orientamento preciso, ad esempio i LED. I diodi permettono il passaggio di corrente solo dal positivo al negativo; i LED, nella vita reale, fortunatamente hanno indicatori quali gamba più lunga o un bordo piatto per indicare il lato positivo, rendendo così più semplice il loro collegamento.

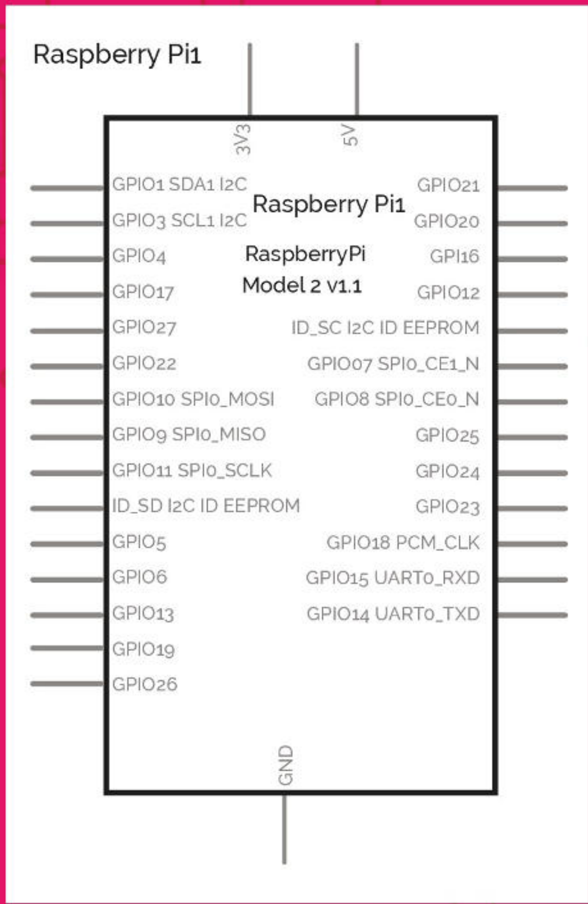
Il Raspberry Pi ed i circuiti electronici

Includere il Raspberry Pi in un circuito è una cosa abbastanza semplice. Nell'utilizzo base, può fornire sia il potenziale positivo che quello negativo o massa, tramite i pin GPIO. Alcuni pin sono appositamente sempre alimentati, per lo più a 3.3V, sempre verso massa. Altri possono invece essere programmati per fornire un segnale ALTO o BASSO oppure per riconoscerlo; nel caso del Raspberry Pi, un segnale ALTO equivale a 3.3V e BASSO è la massa o 0V.

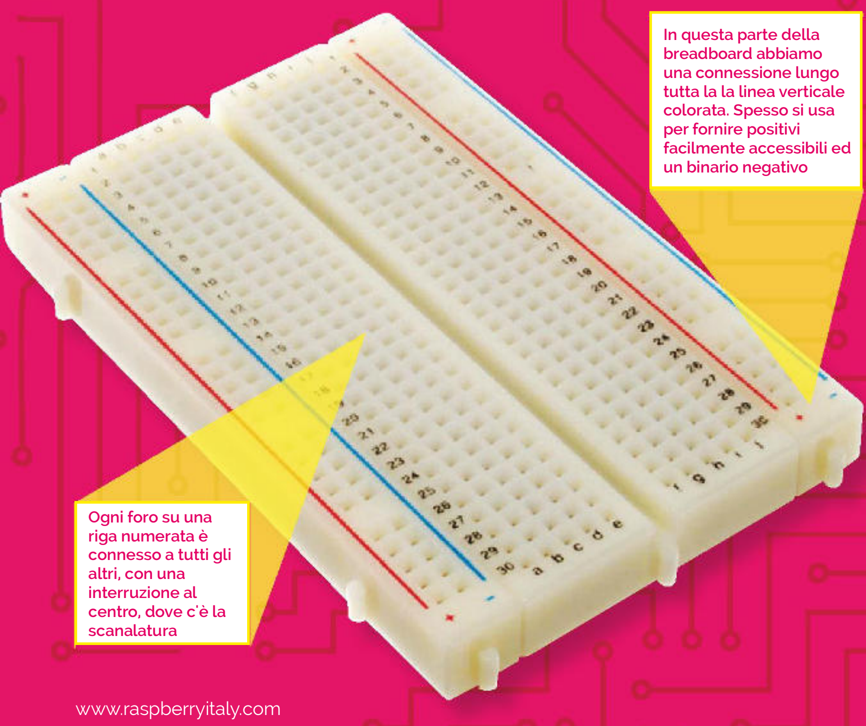
Nell'esempio del LED, puoi collegare un LED direttamente ad un pin da 3.3V e ad un pin di massa e questo si accenderà. Se invece colleghi il terminale positivo del LED ad un pin programmabile del GPIO, si accenderà portando quel pin al valore ALTO(HIGH). Cablare un circuito al Raspberry Pi è discretamente semplice. Per creare il circuito fisico spiegato in questo articolo, abbiamo utilizzato una breadboard. Questo ci consente di inserire i componenti e collegarli tra di loro, senza dover necessariamente creare delle connessioni permanenti.
Questo ti permette di effettuare delle modifiche e di riutilizzare i componenti stessi.

Creare un circuito luce interrotto

Come nostro primo circuito, realizzeremo un interruttore della luce su un breadboard utilizzando Raspberry Pi. Sotto, puoi vedere lo schema circuitale di quello che andremo a costruire; guarda oltre la pagina per una versione illustrata a cui fare riferimento durante la realizzazione del circuito. Le resistenze servono a limitare la corrente sul LED e fungono da ‘pull up’ nel circuito del tasto. Questo porta il pin GPIO a livello BASSO(LOW), quando premuto.

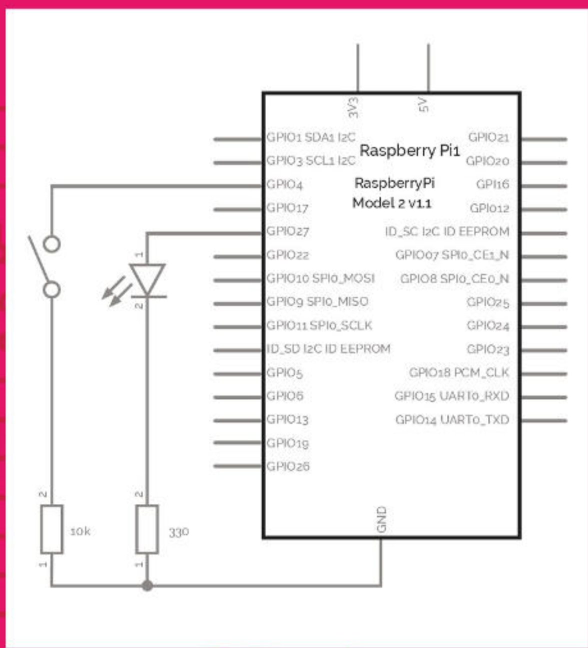


La corrispondenza di numerazione dei singoli pin GPIO è riportata nella pagina seguente.



In questa parte della breadboard abbiamo una connessione lungo tutta la la linea verticale colorata. Spesso si usa per fornire positivi facilmente accessibili ed un binario negativo

Ogni foro su una riga numerata è connesso a tutti gli altri, con una interruzione al centro, dove c'è la scanalatura



GUIDA PER Principianti DI GPIO Zero

GPIO ZERO
leggi la documentazione qui:
magpi.cc/1Od1xtB

Cos'è GPIO Zero e come lo puoi utilizzare per programmare l' elettronica connessa al tuo Raspberry Pi

Una volta che i componenti elettronici sono connessi al Raspberry Pi, devi essere in grado di controllarli. Il Raspberry Pi è particolarmente indicato per essere programmato tramite il linguaggio Python. Il tutto è stato recentemente reso ancora più semplice grazie alla aggiunta di GPIO Zero.

GPIO Zero è stato ideato per semplificare il physical computing, aiutando quindi i principianti che si affacciano al mondo della programmazione. È una libreria Python che si basa sulle esistenti librerie GPIO **RPi.GPIO**, **RPIO**, e **pigpio**. Tuttavia, mentre quelle librerie forniscono una interfaccia ai pin GPIO stessi, GPIO Zero si trova sopra di esse e fornisce una modalità per interfacciarsi ai dispositivi che vengono connessi a quei pin.

Questa variazione semplifica il modo di pensare per il physical computing.
Ipotizziamo di connettere un semplice pulsante tra il pin 4 ed il pin di massa.
Perché venga riconosciuto, dobbiamo sapere che è necessario connetterlo tramite una resistenza, e che lo stato del pin quando si preme il tasto sarà 0.
Ecco come risulterebbe nella classica libreria RPI.GPIO:

```
from RPi import GPIO

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(4, GPIO.IN,
GPIO.PUD_UP)
while GPIO.input(4):
    pass
print("Hai premuto!")
```

Per un perfetto principiante ci sarebbe molto da aggiungere, il che ci porta all'importanza della sperimentazione ed all'insegnamento delle semplici logiche richieste.
Segue il codice equivalente utilizzando GPIO Zero:

```
from gpiozero import
Button

btn = Button(4)
while not btn.is_pressed:
    pass
print("Hai premuto!")
```

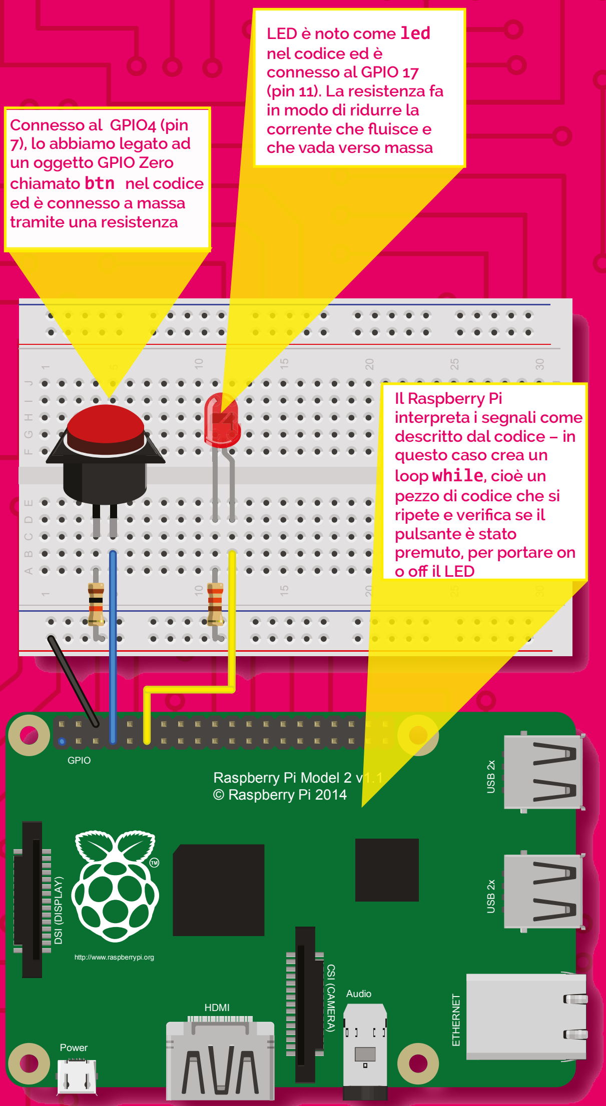
Tutto quel testo che inserivamo alla cieca, senza capire perché era necessario, è ridotto al minimo indispensabile, in modo da avere solo il codice significativo da gestire. Il nome 'GPIO Zero' deriva proprio da questa filosofia che

NUMERI GPIO

Ecco la numerazione dei 40-pin GPIO Raspberry Pi 3, 2, B+, e A+

3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GNV	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPI19	35	36	GPIO16
GPI26	37	38	GPIO20
GND	39	40	GPIO21

GPIO Zero usa la numerazione BCM GPIO al posto del numero del pin – usa questa utile tabella per ricordare la nomenclatura dei pin.



definiremmo 'zero fronzoli', è comparsa per la prima volta nella libreria Pygame Zero di Daniel Pope's.

La logica è molto diretta, senza curiose inversioni dei valori in ingresso.

Accendere un LED

GPIO Zero fornisce diversi metodi per affrontare un problema, così che ogni programmatore possa utilizzare quello che gli risulta più naturale. Qui, abbiamo esteso il nostro script per controllare un LED connesso al pin GPIO17, tramite il pulsante.

Invece di un ciclo di attesa per l'azionamento del pulsante, abbiamo usato un paio di metodi dell'istruzione **wait_for** :

```
from gpiozero import Button, LED

btn = Button(4)
led = LED(17)
while True:
    btn.wait_for_press()
    led.on()
    btn.wait_for_release()
    led.off()
```

Puoi testare questo codice con il circuito nella pagina precedente. Questo metodo funziona per un pulsante, ma diviene equivoco quando se ne usano due. Supponiamo di avere pulsanti connessi ai pin 3 e 4 del GPIO, e vogliamo che il LED si accenda quando si preme uno dei due pulsanti. questo diventa abbastanza difficile con il codice sopra riportato, e quindi ha senso utilizzare invece la programmazione 'event-driven' :

```
from gpiozero import Button, LED
from signal import pause

btn1 = Button(3)
btn2 = Button(4)
led = LED(17)
btn1.when_pressed = led.on
btn1.when_released = led.off
btn2.when_pressed = led.on
btn2.when_released = led.off
pause()
```

Semplice ! Ma c'è un problema. Tenendo premuto un pulsante e premendo e rilasciando l'altro, il LED si spegne a dispetto del fatto che un pulsante è ancora premuto; Pensiamo a come mai un evento viene trascurato e vediamo perché!

Fortunatamente, GPIO Zero fornisce un'altra modalità di programmazione che risolve questo problema consentendo di correlare lo stato del LED allo stato dei pulsanti :

```
from gpiozero import Button, LED
from gpiozero.tools import any_values
from signal import pause

btn1 = Button(3)
btn2 = Button(4)
led = LED(17)
led.source = any_values(btn1.values, btn2.values)
pause()
```

L'esempio utilizza il nuovo modulo **tools** presente in GPIO Zero 1.2, che è progettato per funzionare con le proprietà **source** e **values** sopra utilizzate.

In sostanza, GPIO Zero è eccellente per iniziare con il mondo del physical computing, ma è in grado di svolgere anche cose ben più complesse che quella di attendere che un pulsante venga premuto.

QUALI NOVITÀ' IN GPIO ZERO?

GPIO Zero può usare pulsanti, LED, ronzatori e molti altri componenti. La libreria è in continua espansione.

> INTERFACCIA SERIALE (SPI)

Rilasciata con la versione 1.2.0, è ora presente una implementazione di SPI per specifici dispositivi compatibili per dialogare con il Pi. Questo consente ingressi analogici, convertitori analogico-digitali, ed altri dispositivi avanzati, e ne rende il loro utilizzo molto più semplice.

>EVENTI HOLD

Queste sono variabili utilizzate per programmare un intervallo di tempo di osservazione (ad esempio il tempo che un pulsante deve restare premuto per essere riconosciuto come tale). Questo può essere utile in un progetto in cui non si deve prendere in considerazione un pulsante premuto per errore o per evitare "rimbalzi"

> SOURCE TOOLS

La libreria tools per le proprietà source e values, consente di variare e regolare il modo in cui GPIO Zero gestisce specifici dispositivi e funzioni. Non li tratteremo in questo speciale, ma sono importanti per gli utilizzatori esperti.



PHIL KING

Quando non revisiona e scrive articoli per la rivista *The MagPi*, Phil ama lavorare sui progetti del Pi, incluso il suo nuovo robot a due ruote.

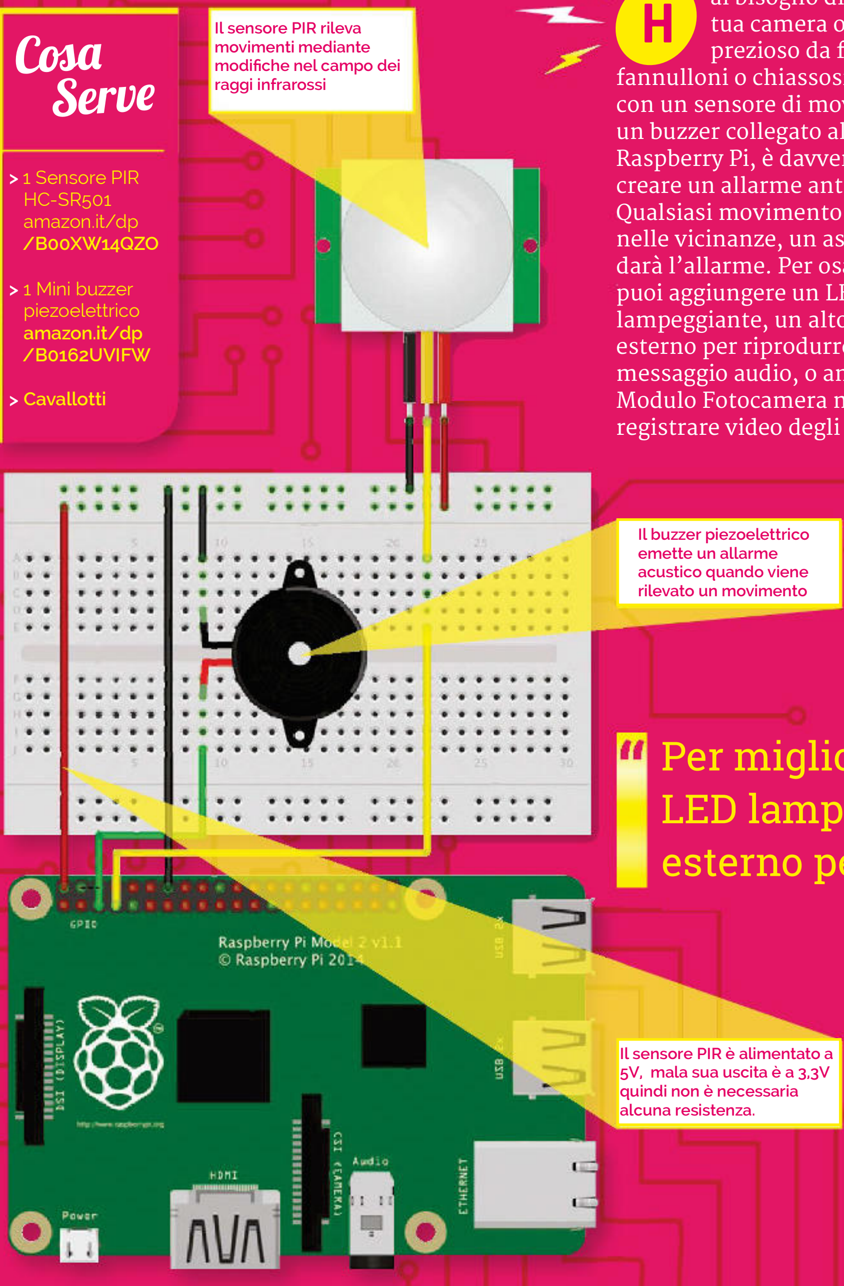
@philking68

Allarme con sensore di movimento

Proteggi le tue cose dai malintenzionati con un allarme che rileva il movimento e suona quando rileva qualcuno

Cosa Serve

- 1 Sensore PIR HC-SR501 amazon.it/dp/B00XW14QZO
- 1 Mini buzzer piezoelettrico amazon.it/dp/B0162UVIFW
- Cavallotti



Hai bisogno di proteggere la tua camera o qualcosa di prezioso da familiari fannulloni o chiassosi? Solamente con un sensore di movimento PIR e un buzzer collegato al tuo Raspberry Pi, è davvero semplice creare un allarme anti intrusione. Qualsiasi movimento sia rilevato nelle vicinanze, un assordante bip darà l'allarme. Per osare di più, puoi aggiungere un LED lampeggiante, un altoparlante esterno per riprodurre un messaggio audio, o anche un Modulo Fotocamera nascosto per registrare video degli intrusi.

>PASSO-01

Collegare il sensore di movimento PIR

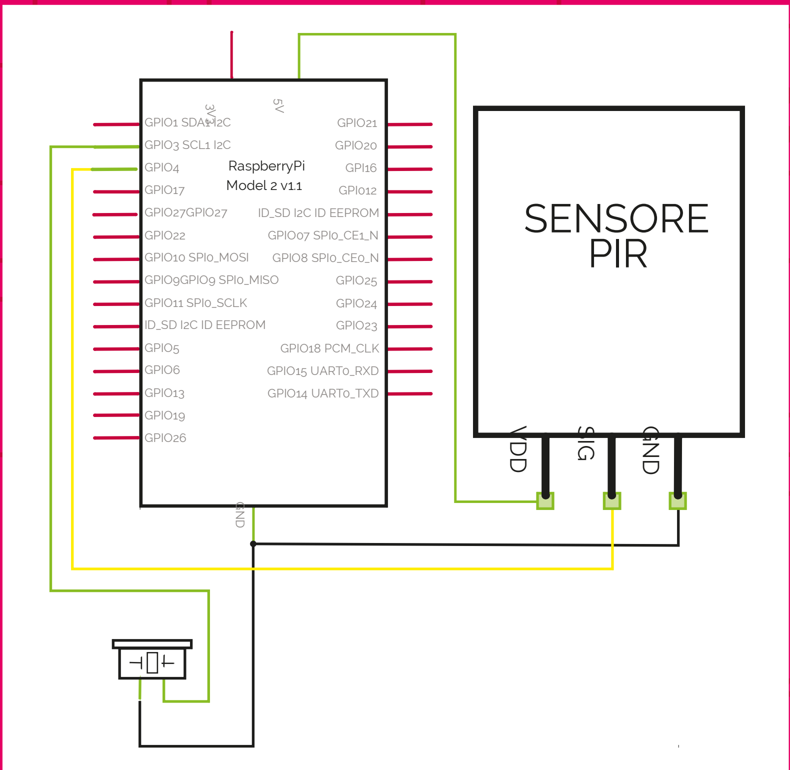
Innanzitutto, abbiamo bisogno di collegare il sensore PIR (a infrarossi passivo) al Pi. Sebbene il sensore potrebbe essere collegato ai pin GPIO direttamente tramite dei cavetti jumper femmina-femmina, lo collegheremo, invece, tramite una breadboard. Il sensore ha tre pin: VCC (tensione di alimentazione), OUT (output), e GND (massa). Utilizza dei connettori maschio-femmina per connettere VCC al binario di alimentazione positiva '+' della breadboard, e GND del sensore al binario di alimentazione negativa '–' . Collega il pin OUT ad una fila numerata, in seguito utilizza un altro cavetto per connettere quella fila al pin 4 GPIO.

>PASSO-02

Collegare il buzzer

In seguito, collegheremo il mini buzzer. Posiziona le sue due gambe a cavallo della scanalatura centrale della breadboard. Tieni conto che la gamba più lunga è il pin positivo; collega una fila numerata al pin 3 del GPIO sul Pi per collegarlo. Connetti una fila alla gamba più corta del buzzer al binario di alimentazione negativo '–' , in seguito connetti quest'ultimo al pin GND sul Pi. Infine, connetti il binario di alimentazione positivo '+' al pin 5V del Pi per alimentare il sensore PIR.

> "Per miglioralo, puoi aggiungere un LED lampeggiante o un altoparlante esterno per riprodurre un messaggio"



PIRalarm.py

Linguaggio
>PYTHON 3

DOWNLOAD:
magpi.cc/1Ynsd0p

```
from gpiozero import
MotionSensor, Buzzer
import time

pir = MotionSensor(4)
bz = Buzzer(3)

print("Attendere la regolazione del PIR")
pir.wait_for_no_motion()

while True:
print("Pronto")
pir.wait_for_motion()
print("Rilevato un movimento!")
bz.beep(0,5, 0,25, 8)
time.sleep(3)
```

>PASSO-03
Lavorare sul codice

All'inizio del programma, importiamo i pratici moduli del **MotionSensor** e **Buzzer** dalla libreria GPIO Zero, ciascuno dei quali contiene numerose funzioni utili; avremo bisogno di alcune di esse per il nostro allarme anti intrusione. Importiamo anche la libreria **time** in modo che possiamo includere un ritardo al ciclo di rilevazione. Successivamente, assegneremo i pin GPIO corrispondenti per il sensore PIR e il buzzer; abbiamo utilizzato rispettivamente i pin 4 e 3 del GPIO, in questo esempio, ma potrai anche utilizzare dei pin alternativi, se preferisci.

>PASSO-04
Impostare il tutto

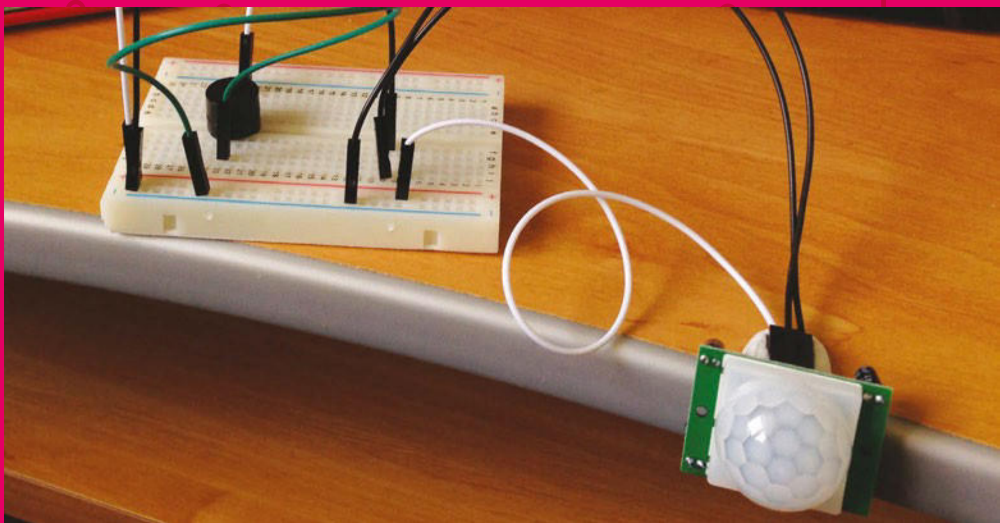
Prima di far partire il nostro ciclo **while** di rilevamento del movimento, facciamo uso della funzione **wait_for_no_motion** della libreria GPIO Zero per attendere che il sensore PIR non rilevi alcun movimento. Questo ti da il tempo di abbandonare l'area in modo che il sensore non percepisca immediatamente la tua presenza e dia l'allarme mentre esegui il codice! Una volta che il sensore PIR non ha rilevato alcun movimento nel suo campo visivo, questo stamperà lo status 'Pronto' sullo schermo e il ciclo di rilevazione del movimento può quindi iniziare.

>PASSO-05
Ciclo di rilevazione movimento

Usiamo **while True:** significa che questo è un ciclo infinito che sarà eseguito ininterrottamente, finché non chiuderai il programma cliccando l'icona 'X' della sua finestra o premendo **CTRL+C** sulla tastiera. Ogni volta che viene rilevato un movimento dal sensore PIR, facciamo suonare il buzzer ripetutamente per otto volte: 0,5 secondi on, 0,25 secondi off, ma puoi modificare la durata. Poi usiamo **time.sleep(3)** per attendere 3 secondi prima di riavviare il ciclo.

>PASSO-06
Regolare la sensibilità

Se pensi che l'allarme scatti troppo facilmente, o non scatti per niente, potresti aver bisogno di regolare la sensibilità del sensore PIR. Lo puoi fare con un piccolo giravite per regolare il potenziometro sul lato, contrassegnato dalla sigla Sx; giralo in senso antiorario per aumentare la sensibilità. L'altro potenziometro, Tx, modifica il ritardo con cui il segnale viene inviato dopo il rilevamento; riteniamo sia meglio girarlo completamente in senso antiorario, per avere il minor ritardo possibile, pari a 1 secondo.



Il programma utilizza un ciclo infinito per rilevare movimenti, facendo suonare il buzzer ogni qualvolta ciò si verifica.



THE HAYLER-GOODALLS

Ozzy, Jasper, e Richard sono mentori alla Fondazione CoderDojo Ham e hanno tenuto un discorso alla festa di compleanno di Rasperry Pi sulle loro avventure Astro Pi.

richardhayler.blogspot.co.uk / **@rdhayler** / **coderdojoham.org**

CREARE UN CPU monitor

Impara come usare un LED RGB con GPIO Zero e a accenderlo a seconda di quanto viene utilizzata la CPU

Il Raspberry Pi 3 è all'incirca dieci volte più potente del modello Raspberry Pi originale – e chi ha utilizzato il Pi negli ultimi quattro anni, può realmente notare la differenza. Anche il salto al Pi 2 è stato abbastanza significativo! Mentre la versione originale del Pi sarebbe quasi sempre al massimo utilizzo di CPU, in maniera sistematica, le versioni del Pi più recenti non fanno altrettanto. Possiamo comunque monitorare l'utilizzo del processore, e con un LED speciale che cambia colore, possiamo creare una spia luminosa che rileva quanto del Pi stiamo utilizzando.

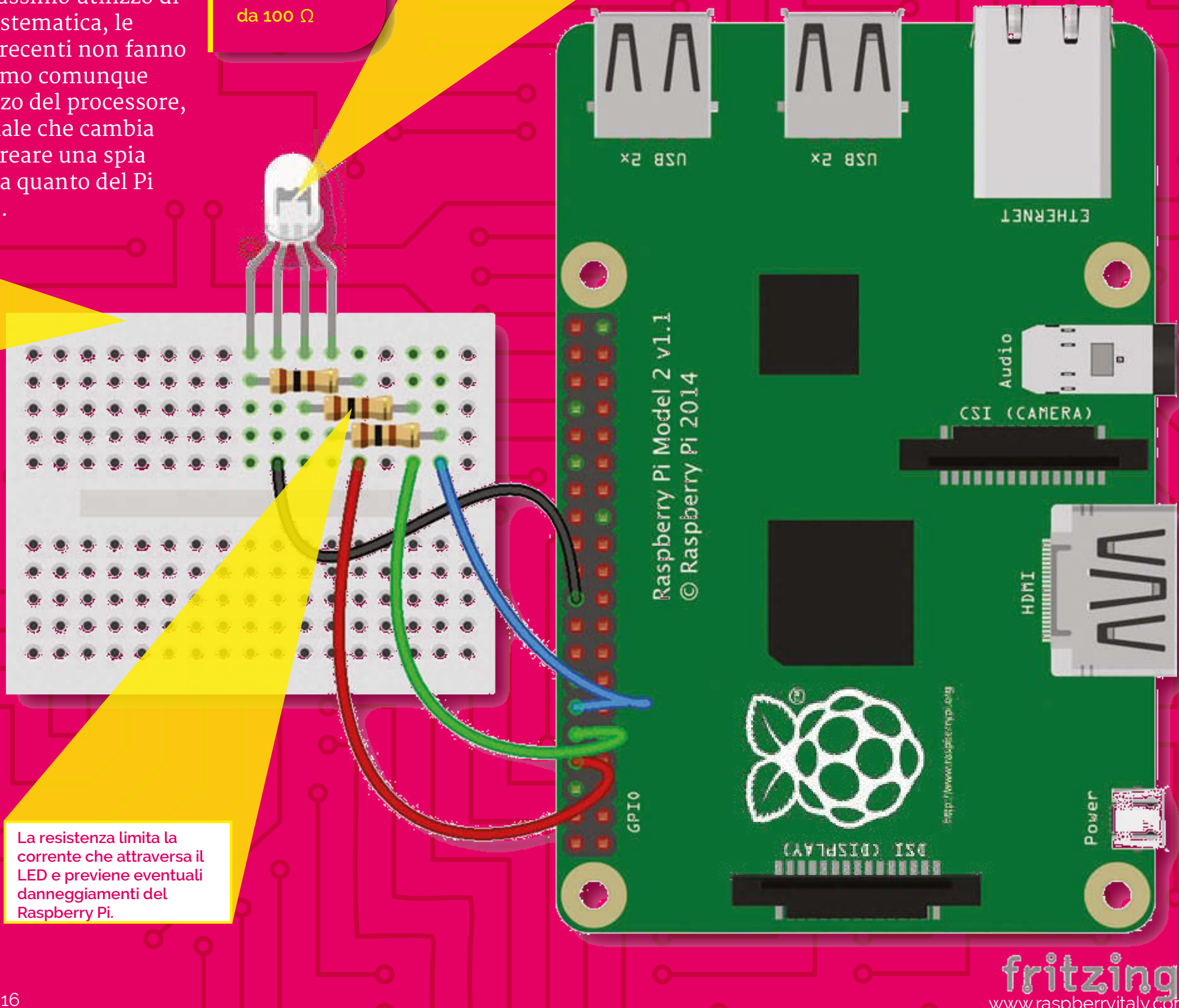
Cosa Serve

- RGB LED
- Cavallotti
- Tre resistenze da 100 Ω

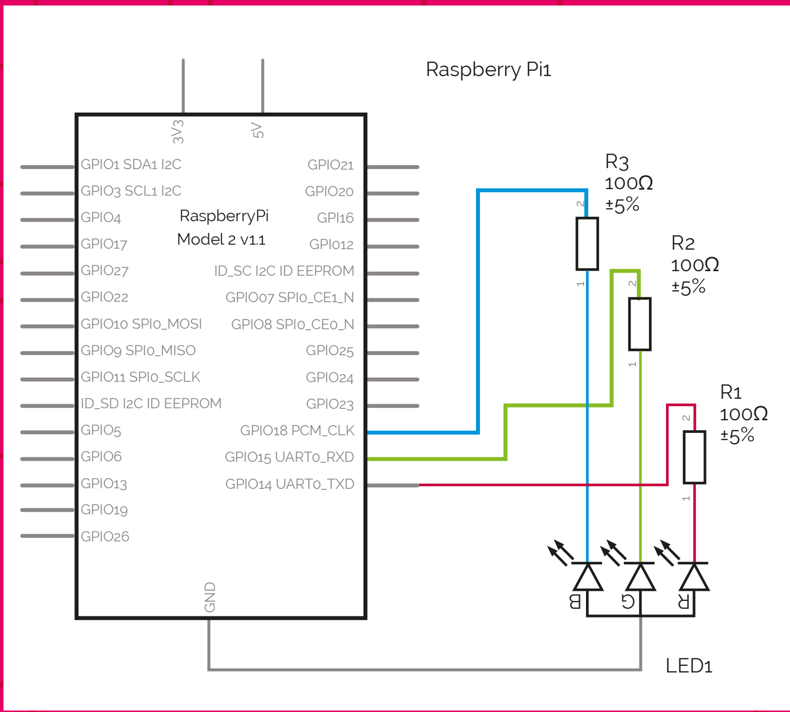
LED GRB a catodo comune. La gamba più lunga è il catodo – connettilo a massa.

Possiamo utilizzare breadboard di qualsiasi dimensione per questo circuito.

Per un LED GRB ad anodo comune, connetti l'anodo (la gamba lunga ancora una volta) al pin 5V (pin 1 del Pi). I pin GPIO sono quindi impostati sul valore basso (LOW) per attivarsi.



La resistenza limita la corrente che attraversa il LED e previene eventuali danneggiamenti del Raspberry Pi.



rgb_cpumon.py

Linguaggio
>PYTHON 3

DOWNLOAD:
magpi.cc/1sy4eC2

```
from gpiozero import
RGBLED
import psutil, colorsys,
time

myled = RGBLED(14,18,15)

while True:
    cpu = psutil.cpu_percent()
    r = cpu / 100.0
    g = (100 - cpu)/100.0
    b = 0
    myled.color = (r,g,b)
    time.sleep(0.1)
```

>PASSO-01
Aggiorna il Pi

Aggiorna il tuo Pi alla versione più recente di Raspbian e assicurati di aver istallato le librerie psutil:

```
sudo pip3 install psutil--upgrade
```

Questo ci permetterà di visualizzare l'utilizzo della CPU del Raspberry Pi come numero percentuale, che potrà poi essere utilizzato nel nostro codice.

>PASSO-02
Scegli il tuo LED RGB

I diodi a emissione luminosa (LED) sono fantastici. Letteralmente. A differenza di una normale lampadina a incandescenza, che ha un filamento incandescente, i LED producono luce esclusivamente dal movimento degli elettroni in un materiale semiconduttore. Un LED RGB ha tre LED colorati inseriti in un corpo unico. Variando la luminosità di ciascuna componente di colore, puoi ottenere una varietà cromatica proprio come se miscelassi della vernice. Esistono due principali tipologie di LED RGB: ad anodo comune e a catodo comune. Noi utilizzeremo quelli a catodo comune.

>PASSO-03
Collegare il LED RGB

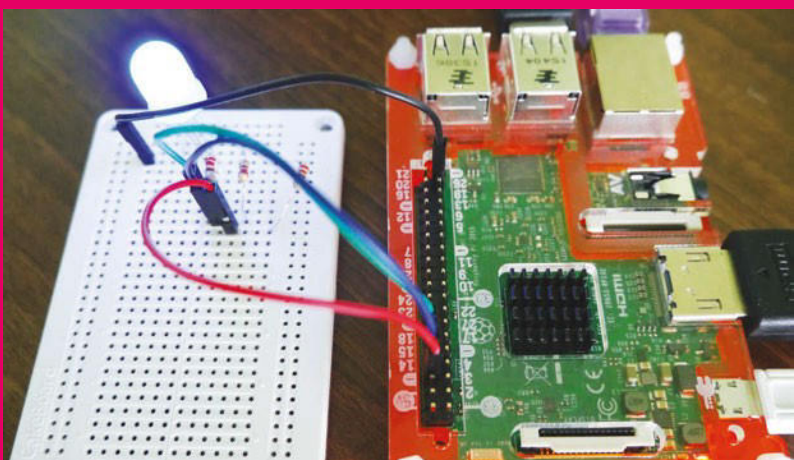
È necessario connettere i LED nel verso corretto. Per un LED RGB a catodo comune, hai a disposizione un solo filo di massa e tre diversi anodi, uno per ciascun colore primario. Per far funzionare questi LED con un Raspberry Pi, connetti ciascun anodo ad un pin GPIO mediante una resistenza per limitare la corrente. Quando uno o alcuni di questi pin è impostato a livello logico ALTO(3.3V), il LED si illuminerà del colore corrispondente. Connetti tutto come mostrato negli schemi.

>PASSO-04
Avviare il codice

Scarica o digita il codice del listato. Il LED dovrebbe accendersi: il suo colore indicherà quanto intensamente stia lavorando la CPU del tuo Pi. Il colore verde indica un livello di utilizzo minimo, varierà più verso il rosso man mano che il carico della CPU andrà a aumentare. Lancia qualche altra applicazione per testarlo. Se possiedi Raspberry Pi originale, Modello B, probabilmente scoprirai che solo lanciare **Minecraft** è sufficiente a far diventare il LED rosso. Se possiedi un PI 3, potresti aver bisogno di eseguire diverse applicazioni in contemporanea per ottenere qualche risultato visibile!

>PASSO-05
Personalizza il tuo progetto

Il codice di esempio utilizza solamente le componenti rossa e verde del LED; il valore del blu è sempre a zero. Potresti cambiare tali impostazioni e creare una differente gradiente di colore (per es. da blu a rosso) o elaborare una fantastica funzione che dia uno specifico valore percentuale a tutti e tre i colori. Divertiti con i colori e magari prendi anche in considerazione altre risorse...



Il CPU monitor in azione. I robusti LED da 10mm costituiscono un indicatore grande e luminoso.



ROB ZWETSLOOT

Riparatore, a volte maker, altre volte cosplayer, e tutto il resto del tempo, capo redattore di ***The MagPi***.

magpi.cc

FARE UN Selfie stick camera

Usa il Modulo Pi Camera, un po' di codice GPIO Zero e fili molto lunghi, per creare un bastone da selfie con pulsante

Molte persone alzano gli occhi al cielo e ne criticano la vanità, quando l'arte del selfie viene espressa, ma tutti noi sappiamo bene che il selfie non è niente di tutto questo. Vestito nuovo? Nuovi occhiali? Il trucco con l'eyeliner oggi è particolarmente simmetrico? Perché non mostrarlo al mondo? È una grande iniezione di fiducia. Prendendo quello che abbiamo fatto finora con GPIO Zero, aggiungendo un tasto e un Modulo Fotocamera Pi, costruiremo il nostro bastone per selfie personalizzato Pi.

>PASSO-01

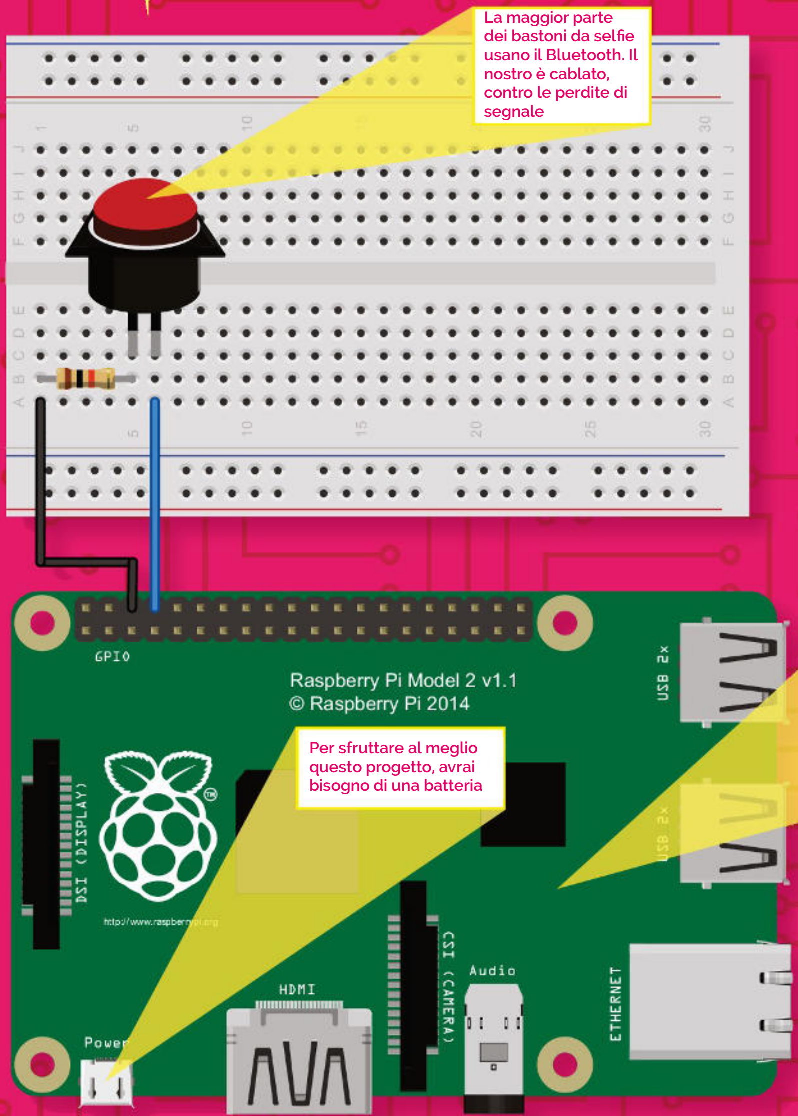
Aggiungere la fotocamera

Un argomento non trattato nell'introduzione all'elettronica, è stato come effettuare l'installazione del Modulo Fotocamera Raspberry, sul Pi. Assicurati che il tuo Pi sia spento e poi trova il connettore per la fotocamera; è il rettangolo bianco tra l' HDMI e le porte audio, sulla Pi 2 e 3. Solleva delicatamente la slitta, e quindi inserisci il cavo a nastro della fotocamera Pi nella fessura con i connettori argentati dal lato del connettore della porta HDMI. Spingi ora la slitta nuovamente verso il basso.

>PASSO-02

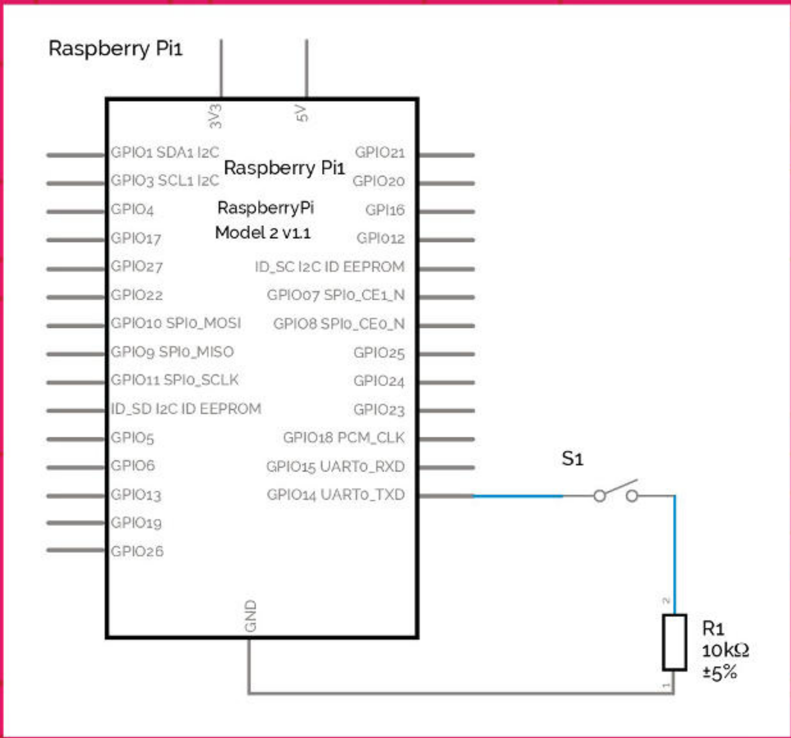
Camera software

Per far funzionare la camera sul Raspberry Pi e in Python, abbiamo



Cosa Serve

- Un contenitore per Raspberry Pi con camera
- Raspberry Pi Camera Module amazon.it/dp/B01ER2SKFS
- Un pulsante e alcuni fili molto lunghi
- Un bastone sottile in metallo o qualsiasi altra cosa per montarci su il case del Pi



selfie.py

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/Selfie-Camera

```
from gpiozero import
Button
from datetime import
datetime
import picamera
import time

btn = Button(14)
pc = picamera.PiCamera()

def picture():
    timestamp = datetime.now()
    pc.capture('pic'+str(
timestamp)+'.jpg')
    time.sleep(1)
while True:
    btn.wait_for_press
    picture()
```

bisogno di fare un paio di cose. Primo, andare nel menu Programmi e aprire l'interfaccia per la configurazione di Raspberry Pi. Vai alla scheda Interfacce e attiva la fotocamera ,se non è già stato fatto. Successivamente, apri un finestra del terminale e installa il modulo Python picamera, con questo comando:

```
sudo apt-get install
python3-picamera
```

>PASSO-03
Assemblaggio

Fotocamera e Pi devono essere collegati, la cosa più furba è quindi procurarsi un cavo a nastro sufficientemente lungo per collegare il Modulo Fotocamera al Pi. Fissa il Pi in un case a una estremità del bastone, come meglio credi (colla, Blu-Tack, velcro, ecc) e quindi collegaci il tuo bottone.

Noi abbiamo utilizzato gli stessi cavallotti mostrati nella introduzione all'elettronica, ma abbiamo aggiunto un lungo tratto di filo fra la loro estremità e il pulsante che abbiamo usato.

>PASSO-04
Aggiungere il codice

Scarica o digita il codice del listato. Puoi fare una rapida prova eseguendo lo script e poi scollegare i cavi HDMI e USB dal Pi. Premendo il pulsante inizierà a scattare foto, ma dovrai fare un po' di pratica prima di riuscire a inserirti nella inquadratura. posiziona ora tutto al suo posto, e dai un'occhiata al codice in modo che possiamo spiegarti quello che stai facendo con esso.

>PASSO-05
GPIO Zero e Python

L'uso di GPIO Zero in questo codice è abbastanza semplice e lo hai già visto nelle pagine precedenti. Noi abbiamo impostato il pulsante su GPIO 14 (pin 8) e lo utilizziamo esclusivamente come interrupt su un ciclo **while** che controlla se viene premuto il pulsante. Abbiamo anche utilizzato **datetime** e **time** per, rispettivamente, creare un timestamp sull'immagine e ottenere una piccola pausa nell' esecuzione del codice. Questo rende più semplice organizzare le foto e evita di catturare troppe immagini con una sola pressione del tasto.

>PASSO-06
Libreria picamera

Ci sono molti modi per utilizzare questa libreria, noi useremo il più semplice. Abbiamo creato **pc** come variabile fotocamera, in maniera similare a quando si imposta qualsiasi altra variabile GPIO Zero. Quando vogliamo scattare una foto, viene utilizzato il comando **capture**, e facciamo in modo che l'immagine venga nominata con il timestamp catturato. È semplice, ma efficace, e non mancherà di tenerti a scattare foto finché ci sarà ancora spazio sulla scheda SD.



l nostro bastone per selfie di prova, è molto casereccio, ma puoi usare qualsiasi cosa, su cui si può fissare il Pi, e che sia abbastanza lunga



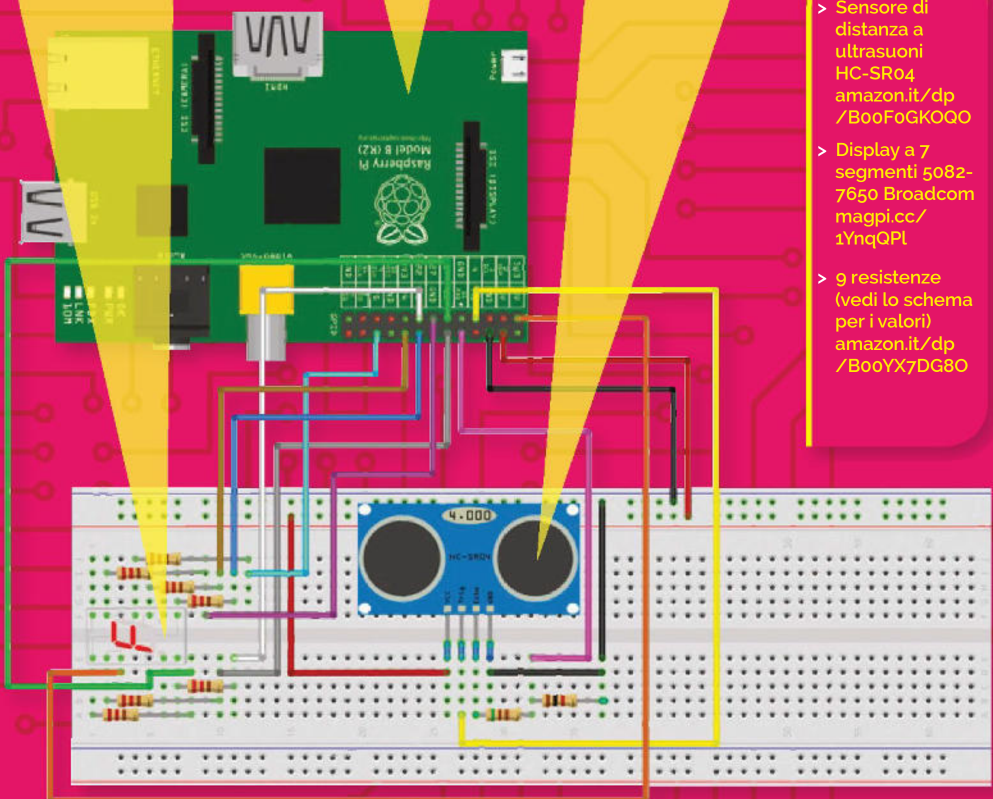
MIKE COOK

Veterano della vecchia guardia della rivista, autore della serie Body Build, e co-autore di tre libri su Raspberry Pi.

thebox.myzen.co.uk

Fare un telmetro

Metti insieme un sensore a ultrasuoni e un display a sette segmenti per misurare le distanze



Cosa Serve

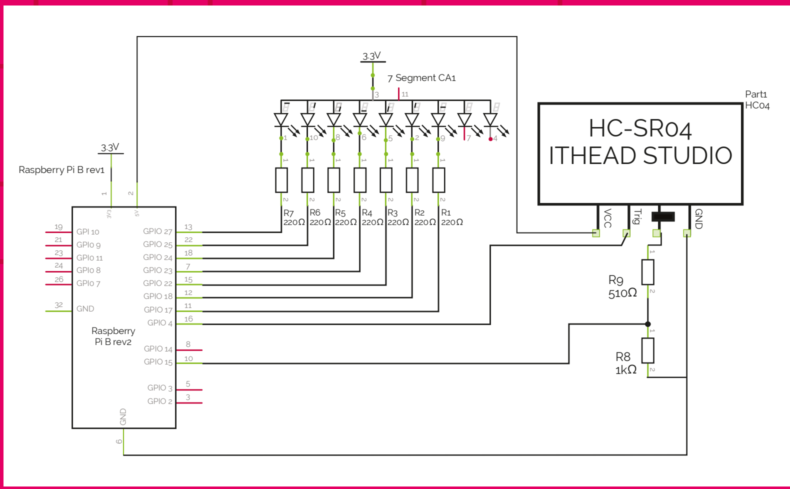
- Sensore di distanza a ultrasuoni HC-SR04 amazon.it/dp/B00F0GKOQO
- Display a 7 segmenti 5082-7650 Broadcom magpi.cc/1YnqQPl
- 9 resistenze (vedi lo schema per i valori) amazon.it/dp/B00YX7DG8O

Il sensore di distanza ad ultrasuoni HC-SR04 è uno dei preferiti dai maker che fanno robot col Pi. Funziona emettendo un suono ultrasonico e cronometrando il tempo impiegato dal suono a rimbalzare sugli oggetti e tornare indietro. Questo tempo viene poi convertito in una distanza che può essere visualizzata su un display a sette segmenti. In questo articolo acquisirai le competenze necessarie per gestire ingressi e uscite. Vedrai anche come utilizzare il display a sette segmenti, piuttosto figo, nel suo genere retrò.

>PASSO-01

Accendere il display

Il display a sette segmenti è formato da LED, a ogni segmento corrisponde un LED. Tutti gli anodi (terminali positivi) sono collegati tra loro. Devono essere connessi all'alimentazione da 3.3V. I catodi (terminali negativi) devono essere invece collegati a una resistenza per limitare la corrente che attraversa i LED, mentre l'altra estremità della resistenza va ad un pin GPIO. Per accendere il LED, tutto quello che dovrai fare è impostare l'uscita del GPIO in modo che sia a livello logico BASSO (0V), così da completare il circuito e far fluire la corrente.



>PASSO-02
Genrare un disegno con sette segmenti

Il display è costituito da quattro barre o segmenti che possono essere illuminati. Scegliendo i segmenti da illuminare, puoi visualizzare un numero da 0 a 15, anche se per farlo devi ricorrere anche alle lettere (nota anche come numerazione esadecimale). Vi sono, in effetti, 128 diversi disegni che puoi generare, ma la maggior parte sono senza senso. Una lista chiamata **seg** definisce quale pin è connesso a quale segmento, e un'altra lista chiamata **segmentPattern** definisce lo schema LED per ogni numero.

>PASSO-03
Visualizzare numeri

La funzione display imposta i segmenti per visualizzare qualsiasi numero a una cifra che gli viene passato. Inizialmente, imposta tutti i segmenti spenti, e poi, se il numero è inferiore a 16, scorre gli elementi nella lista **segmentPattern** per quel numero, e accende i segmenti appropriati. Nota che possiamo ancora usare on e off, anche se i segmenti non sono alimentati dai singoli pin GPIO, in quanto i LED sono stati dichiarati a GPIO Zero come **active_high = False**

>PASSO-04
Il sensore di distanza

Il sensore di distanza HC-SR04 emette un impulso in uscita che il Pi tenta di misurare. La libreria GPIO Zero misura questo impulso e lo converte in una distanza, restituendo un numero in virgola mobile, con un massimo di 1 metro. Abbiamo poi moltiplichiamo questo numero per 10, per ottenere una misura in decimetri. Poi, lo abbiamo convertito in un numero intero, per sbarazzarci delle cifre decimali della misura, e poterla così visualizzare sul display a una cifra.

>PASSO-05
Assemblare il progetto

Per la nostra realizzazione, abbiamo utilizzato una piccola breadboard shield di Dtronixs. Questa ha permesso una disposizione dei componenti molto più compatta rispetto a una breadboard tradizionale, che naturalmente, è comunque possibile utilizzare. Siccome la HC-SR04 utilizza una alimentazione di 5V, l'impulso che dobbiamo misurare sarà anch'esso di 5V nominali. Pertanto, deve essere ridotto a 3,3V utilizzando un partitore di tensione costituito da una resistenza di 512Ω e una da 1kΩ.



Il progetto in azione; il Pi sta misurando quanto è lontano dalla scatola del Raspberry Pi 3.

>PASSO-06
Usare il sensore

La distanza dall'oggetto riflettente viene aggiornata ogni 0,8 secondi. Se questa è maggiore di un metro, allora il display sarà vuoto. Uno 0 indica invece che l'oggetto è a meno di 10 cm di distanza. Non toccare il sensore, altrimenti le sue letture saranno falsate. Inoltre, in quanto ha una emissione piuttosto larga, è possibile avere dei riflessi laterali. Se più oggetti sono nel campo di azione, allora viene restituita la distanza del più vicino.

displayDistance.py

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/1NqJjmV

```
# mostra la distanza in
decimetri sul display
a sette segmenti
from gpiozero import LED
from gpiozero import DistanceSensor
import time

seg = [LED(27,active_
high=False),LED(25,active_
high=False),LED(24,active_high=False),
      LED(23,active_
high=False),LED(22,active_
high=False),LED(18,active_high=False),
      LED(17,active_high=False)]

segmentPattern = [[0,1,2,3,4,5],[1,2],[0,
1,6,4,3],[0,1,2,3,6],[1,2,5,6],[0,2,3,5,6
], #0 to 5
                 [0,2,3,4,5,6],[0,1,2],[
0,1,2,3,4,5,6],[0,1,2,5,6],[0,1,2,4,5,6]
, #6 to A
                 [2,3,4,5,6],[0,3,4,5],[1
,2,3,4,6],[0,3,4,5,6],[0,4,5,6] ] #B to F

sensor = DistanceSensor(15,4)

def main() :
   print(
"Mostra distanza sul display a 7 seg")
   while 1:
       distance = sensor.distance * 10 #distanza
in decimetri
       print("distanza",distance)
       if distance >= 10.0:
          distance = 16.0
       display(int(distance))
       time.sleep(0.8)

def display(number):
    for i in range(0,7):
       seg[i].off()
    if number < 16:
       for i in range(0,len(
segmentPattern[number])):
           seg[segmentPattern[
number][i]].on()

# logica programma principale:
if __name__ == '__main__':
    main()
```

ROB ZWETSLOOT

Riparatore, a volte maker, altre volte cosplayer, e tutto il resto del tempo, capo redattore di *The MagPi*.

magpi.cc

REALIZZA UNA Internet radio

Utilizzando caratteristiche evolute di GPIO Zero, crea una radio internet che utilizza una manopola per cambiare stazione, proprio come una radio vecchio stile

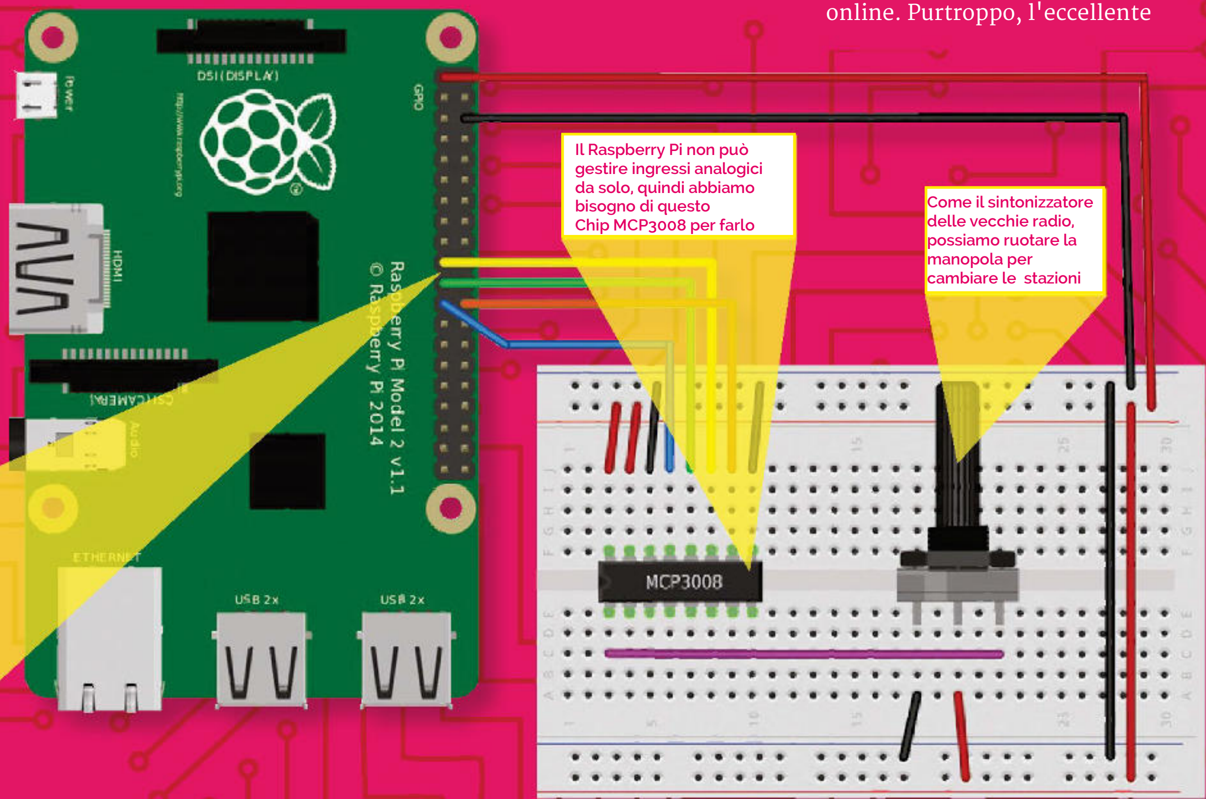
Questo è un progetto carino. Il codice originale di questo progetto è stato sviluppato dal boss di The MagPi Russell Barnes molto tempo prima che GPIO Zero vedesse la luce. Il codice era lungo oltre 100 linee e ti permetteva di variare il volume e di selezionare la stazione radio tra cinque diverse stazioni, utilizzando delle resistenze variabili o dei potenziometri. Ora, grazie all'ultima versione di GPIO Zero, abbiamo ridotto il codice fino a 21 linee, anche se questo ha comportato l'abbandono del controllo del volume, per il momento. Ti mostreremo come abbiamo fatto.

>PASSO-01
Software multimediale

Prima di fare qualsiasi cosa con il codice, dobbiamo assicurarci di poter effettivamente mandare in riproduzione la stazione radio. Useremo gli stream M3U online per farlo, che sono disponibili per molte stazioni radio digitali o online. Purtroppo, l'eccellente

Cosa Serve

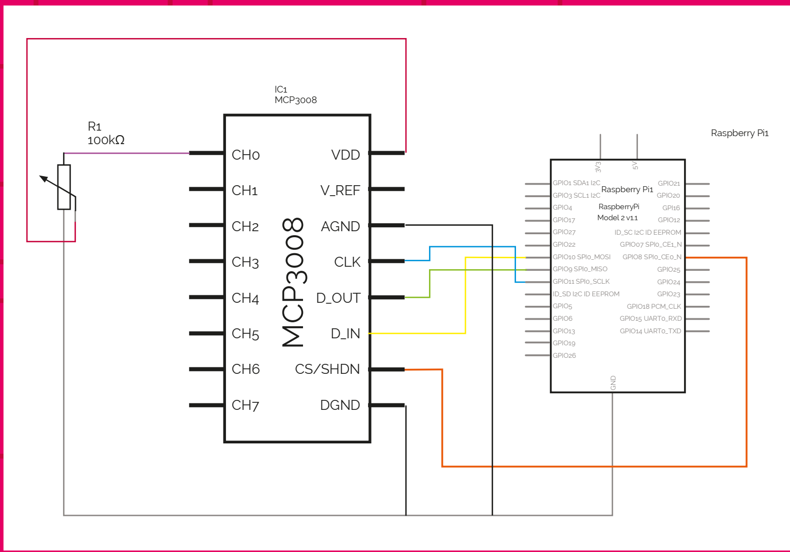
- Convertitore analogico-digitale MCP3008
- Mplayer
- Potenziometro o resistenza variabile
- Uscita audio



Il Raspberry Pi non può gestire ingressi analogici da solo, quindi abbiamo bisogno di questo Chip MCP3008 per farlo

Come il sintonizzatore delle vecchie radio, possiamo ruotare la manopola per cambiare le stazioni

Il MCP3008 è collegato a diversi pin specifici del Pi – questi pin esistono su tutti i modelli



internetradio.py

Language
>PYTHON 3

DOWNLOAD:
magpi.cc/1OG2Y9i

```python
from gpiozero import
MCP3008
import time
import os

station_dial = MCP3008(0)

Magic = "http://tx.whatson.com/icecast.
php?i=magic1054.mp3.m3u"
Radio1 = "http://www.listenlive.eu/
bbcradio1.m3u"

def change_station(station):
    os.system("killall mplayer")
    os.system("mplayer -playlist " +
station + " &")

while True:
    if station_dial.value >= 0.5:
        station = Magic
        change_station(station)
    elif station_dial.value < 0.5:
        station = Radio1
        change_station(station)
    time.sleep(0.1)
```

omxplayer - che è il lettore multimediale predefinito di Raspberry Pi - non può farlo, quindi abbiamo la necessità di installare mplayer dal terminale con:

```
sudo apt-get install
mplayer
```

>PASSO-02
Costruire la radio

Il circuito radio è costituito da due parti principali: il chip MCP3008 e la resistenza variabile. Il chip ci consente di leggere l'output della resistenza variabile sul Pi. Assicurati, quando lo andrai a inserire sulla breadboard, che sia posizionato a cavallo della piccola scanalatura che ne attraversa il centro; in questo modo i pin non sono cortocircuitati tra di loro. Presta attenzione durante il cablaggio: assicurati che la piccola tacca posta sopra il chip sia orientata dal lato corretto, come mostrato nel diagramma.

>PASSO-03
Collegare un potenziometro

È necessario che identifichi i pin sulla resistenza variabile / potenziometro. Comprendono una massa, un ingresso e una uscita, che sono collegati ai piedini 1 , 2 e 3 in questo ordine. Nel nostro circuito, l'ingresso è semplicemente il positivo 3V3 del Raspberry Pi. L'uscita è poi collegata ad uno degli ingressi del MCP3008, che in questo caso si chiama canale 0. Girando il potenziometro, cambierà la resistenza e quindi la corrente che circola nel canale 0.

>PASSO-04
Aggiungere il codice

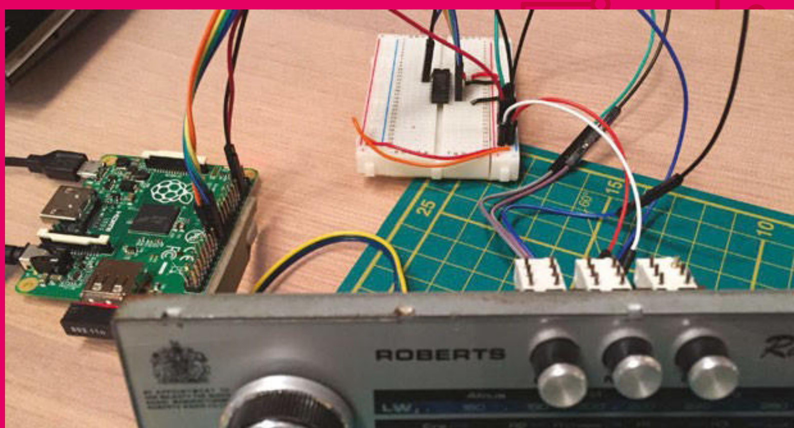
Digita o scarica il codice per la radio. La magia di questo codice è che in precedenza erano necessarie molte righe di codice dedicate a garantire che il chip MCP3008 venisse letto correttamente. Ora, fintanto che il chip è collegato correttamente, dobbiamo solo dire a GPIO Zero quale canale ci interessa con la linea **station_dial = MCP3008(0)**. Questa è una delle più recenti aggiunte a GPIO zero, come parte dell'aggiornamento v1.2.

>PASSO-05
Il codice

Il valore del potenziometro, riportato dal chip, sarà un numero tra 0 e 1. Siccome abbiamo solo due stazioni, abbiamo fatto in modo che ogni direzione di rotazione della manopola corrisponda a una stazione diversa. Puoi aggiungere più stazioni e più selezioni, come puoi intuire. Abbiamo utilizzato anche il modulo **os** in questo codice; il che ci permette di inviare comandi direttamente al terminale, e in questo caso ci permette di lanciare mplayer.

>PASSO-06
Miglioramenti

Questo codice è molto adattabile, se hai voglia di modificarlo. Cambiare le stazioni radio o aggiungerne di supplementari è molto semplice, e potrebbe anche includere facilmente qualche sorgente di tipo diverso o una uscita audio migliore. Puoi anche aggiungere un altro potenziometro e ripristinare il controllo del volume, rendendolo così una vera e propria radio.



Perché non provare a mettere il prodotto finito all'interno di un contenitore di una vecchia radio? I comandi sul frontale funzionano in modo analogo